Autoencoder based Anomaly Detection for SCADA Networks

Sajid Nazir
School of Computing, Engineering and Built Environment,
Glasgow Caledonian University, Glasgow, G4 0BA, UK
sajid.nazir@gcu.ac.uk

Shushma Patel
School of Engineering, London South Bank University,
London, SE1 0AA, UK
shushma@lsbu.ac.uk

Dilip Patel
School of Engineering, London South Bank University,
London, SE1 0AA, UK

ABSTRACT

Supervisory Control and Data Acquisition (SCADA) systems are industrial control systems that are used to monitor critical infrastructures such as airports, transport, health and public services of national importance. These are cyber physical systems, which are increasingly integrated with networks and Internet of Things devices to provide benefits such as timely operational feedback and visibility. However, this results in a larger attack surface for cyber threats, making it important to identify and thwart cyber-attacks by detecting anomalous network traffic patterns. Compared to other techniques, as well as detecting known attack patterns, machine learning can also detect new and evolving threats. Autoencoders are a type of neural network that generates a compressed representation of its input data and through reconstruction loss of inputs can help identify anomalous data. In this paper we propose the use of autoencoders for unsupervised anomaly based intrusion detection using an appropriate differentiating threshold from the loss distribution and demonstrate improvements in results compared to other techniques for SCADA gas pipeline dataset.

Keywords: anomaly detection, SCADA, clustering, classification, IoT, neural networks, intrusion detection, machine learning, autoencoders

INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems are cyber physical systems distributed over a large geographical area and if compromised, can severely impact public health and safety. A SCADA system (Figure 1) comprises of a layered architecture from lowest level simple sensors and actuator field devices through Programming Logic Controllers (PLC) and Remote Terminal Units (RTU) over a communications networks to the highest layer comprising SCADA servers and Human Machine Interface (HMI).

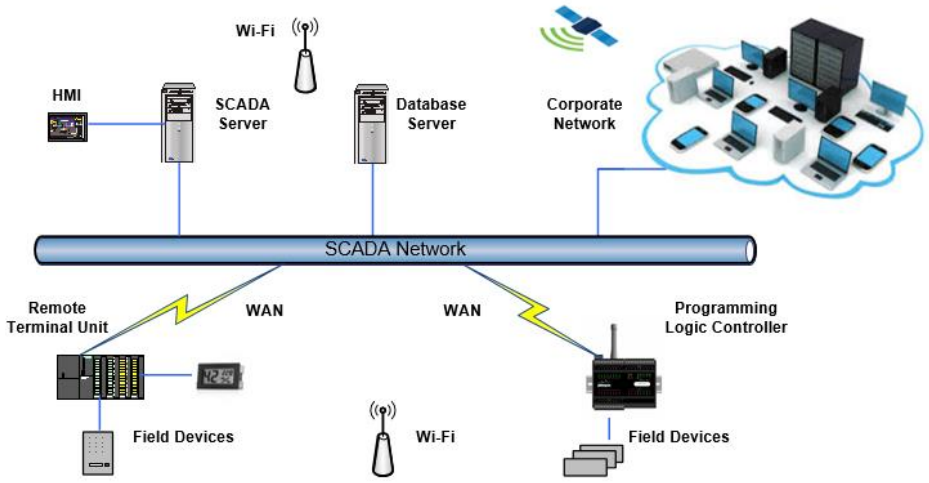

Figure 1. A geographically distributed SCADA system interconnected through Internet and wireless communications

Historically, SCADA systems were designed as on-site networked systems which were not accessible over the Internet. Thus cyber intrusions required physical access to the system, for example, as in the case for Stuxnet (Langner, 2011). Over time, SCADA systems have increasingly been connected to the Internet and a natural progression to Internet of Things (IoT) connectivity has taken place. Internet accessibility provides many benefits, better communications protocols, cost effectiveness and remote access; however networked SCADA systems get exposed to a large number of cyber threats (Genge et al., 2012). SCADA systems and protocols were not designed with off-site network connectivity in mind as security was not a serious concern for an isolated and secure system. However, with interconnectivity and open standards serious vulnerabilities in SCADA system have been observed (Erol-Kantarci, & Mouftah, 2013). The sensors and actuators in modern SCADA systems can communicate over a variety of communications media, such as WiFi, cellular and Bluetooth. Thus SCADA systems comprise of many old as well as new communications technologies, potentially providing many entry points for an attack from around the globe (Nazir, et al., 2018).

Vulnerabilities in the communications protocols can be exploited to launch cyber-attacks. Internet and cellular network connectivity have amplified the threat (Zhu et al., 2011), as attackers can exploit known security loopholes in open standards to gain access to SCADA systems (Igure et al., 2006). The widespread availability of free protocol information, increased general technology awareness and the current global security situation has made such attacks easier and more likely to be launched (Nazir, et al., 2018). Thus these systems have come to the attention of malicious users as evidenced by the steadily increasing number of attacks over recent years (Cyber Security Breaches Survey, 2017). One way to counter the cyber threats is by learning to identify an attack instance in the network traffic.

SCADA systems are event driven and under normal operations most of the commands and responses are time or event triggered, making it possible to use security and monitoring approaches unlike in open environments (Mantere, et al., 2013). The Intrusion Detection Systems (IDS) research requires realistic datasets for normal and attack scenarios which are generally not available for training and testing the algorithms (Buczak, & Guven, 2016). A network traffic data log for gas pipeline was created by (Morris et al., 2015) providing both normal and attack operations. The regular command-response patterns are repetitive, which make them suitable for detecting anomalous behaviours (Turnipseed, 2015).

Machine learning techniques can analyse and process data to isolate anomalous instances, which signal malicious behaviour, thereby making automated machine learning techniques more appropriate and efficient compared to human analysts (Jiang & Yasakethu, 2013). A survey of machine learning and data mining for IDS is outlined by Buczak, & Guven (2016) and provides details of the datasets. Although supervised techniques have been applied for intrusion detection, they are not very effective because they require labelled datasets. Attack data is not available in sufficient quantities to train supervised machine learning techniques, compared to normal operation data for SCADA systems. However unsupervised techniques work better to identify novel threats as they do not require labelled data. Also, supervised techniques, even after training, are less capable of detecting unseen patterns. Thus in unsupervised techniques, the normal traffic features can be learned and an unseen pattern can be detected as an intrusion based only on its error or distance from the normal dataset. An unsupervised machine learning technique gaining prominence for anomaly detection is autoencoder (Aygun, & Yavuz, 2017; Yousefi-Azar, et al., 2017).

The main contributions of this paper, for investigating anomaly detection are:

- Feature engineering the dataset for selecting and extracting the most relevant features using Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) for subsequent binary and multi-class anomaly detection
- Use of autoencoders to detect unseen attack patterns based on the reconstruction loss through selecting an appropriate differentiating threshold from the reconstruction loss distribution

RELATED WORK

Machine learning techniques have been applied to identify previously unseen threat patterns which uncover features and connections hitherto unknown or unobserved by other techniques in SCADA cyber security. The research applying machine learning techniques using publicly available SCADA datasets and some of the related techniques on public datasets are outlined in Table 1. One Class Support Vector Machine (OCSVM) for automated anomaly detection (Schuster, et al., 2015) from SCADA telecommunications data was used by (Jiang & Yasakethu, 2013; Maglaras, & Jiang, 2014). They proposed clustering the anomalies into different types to generate a corresponding alarm.

An unsupervised anomaly based detection scheme was proposed (Almalawi, et al., 2014) for a water distribution system and dataset from an urban waste water treatment plant (Lichman, 2013) by detecting inconsistent states and proximity based detection rules. Man-in-the-middle (MITM) attacks on Modbus/TCP were investigated and the scheme was compared by (Almalawi, et al., 2014) to supervised and semi-supervised schemes to provide better results.

Table 1. Techniques Applied for Intrusion Detection in SCADA Systems

Dataset	Technique	Reference
Gas Pipeline	MLP with GWO	(Mansouri, et al., 2017)
Gas Pipeline	K-means, Naïve Bayesian, PCA-SVD, GMM	(Shirazi et al., 2016)
Gas pipeline	LSTM	(Feng et al., 2017)
Water Distribution System (DUWWTP)	KNN, K-means	(Almalawi, et al., 2014)
DUWWTP, Gas pipeline	SVDD, PCA	(Nader et al., 2014)
Network trace	OCSVM, K-means	(Maglaras, & Jiang, 2014)
CERT Insider Threat	RNN	(Tuor et al., 2017)

Intrusion detection that complements the normal IDS is proposed in (Nader, et al., 2014). The paper investigates intrusion detection using classification techniques of Support Vector Data Description (SVDD) and the Kernel Principal Component Analysis (KPCA), showing better error detection and a reduced false alarm rate. The selected algorithms provided faster and better results on selected datasets (Lichman, 2013; Beaver, et al., 2013). A comparison of 24 machine learning algorithms is provided (Mansouri, et. al., 2017) for anomaly detection in a gas distribution network (Beaver, et al., 2013) and dimensionality reduction techniques for improving accuracy were also used. In addition, they proposed a new algorithm for anomaly detection in SCADA. 1% of the data from 97019 entries of a gas dataset was used. The gas pipeline dataset (Beaver, et al., 2013) was used by (Shirazi, et al., 2016) to implement various anomaly detection techniques to detect attacks. They evaluated K-means, Guassian Mixture Model (GMM), Principal Component Analysis-Singular Value Decomposition (PCA-SVD) and Naïve Bayesian (NB) techniques in supervised mode and Principal Component Analysis (PCA) in unsupervised mode for anomaly detection on (Beaver, et al., 2013) dataset. They used the whole dataset with its seven attack classes, and also used the normal data with one of the seven attack classes thus yielding 8 trace sets utilizing a selected subset (30%). However, the current research focus has shifted to the application of neural networks for intrusion detection in SCADA cyber security, as these techniques provide promising results in other complex problems such as Natural Language Processing (NLP) and image annotation (Liu, et. al., 2017).

A deep learning framework based on stacked autoencoder was proposed for attack detection and classification in smart grids (Wilson et al., 2018). The proposed framework was used for unsupervised feature learning in complex security scenarios. Deep denoising autoencoders were used to propose a reconstruction scheme to mitigate the impact of covert cyber-attacks in smart grids (Ahmed et al., 2019). The results identified a low error ratio, compared to other schemes. A maintenance information based method is proposed for anomaly detection in SCADA data from wind turbine (Lutz et al., 2020). An autoencoder based method is proposed for unsupervised anomaly detection and performance benchmarking in building energy data (Fan et al, 2018).

A signature database was created (Feng, et al., 2017) with Bloom filter and used as the next stage by Long Short Term Memory (LSTM) softmax classifier for anomaly detection on gas pipeline dataset (Beaver, et al., 2013). The classification is based on packet level detection centred on features and a time series detection using previously seen packets. A stacked LSTM was trained with unlabelled data and the training made use of probabilistic noise labelled packets to reduce the False Positive (FP) rate. The anomalous packets used for the training and validation were manually removed although the testing data contained anomalous packets. For the Bloom filter some of the features that were correlated were clustered and augmented with a discrete value to help reduce FP to 0.03. The LSTM model was trained both with and without the added noise for 50 epochs. Feng, et al.(2017) provide comparison of their

proposed techniques to other techniques such as PCA-SVD, GMM and Bayesian networks (BN) and showed improved results.

In comparison, in this paper, we consider the full dataset, using Recursive Feature Elimination (RFE) and PCA on the data set to select and transform the important features. We focus on the application of autoencoders to detect anomalies through binary classification but also provide multi-class classification results using other techniques for comparison with published results.

BACKGROUND

Anomaly Based Intrusion Detection

An anomaly is a value or outcome that deviates from the expected or normal value (Alla, 2019). Anomaly detection identifies an outlier or deviant value from the normal traffic (Igure et al., 2006; Görnitz, et al., 2013). Intrusion detection systems are based on signature or rule based threat detection. However, such schemes do not provide protection against unseen threat patterns.

A survey of IDS technologies for SCADA systems is provided by (Heenan, & Moradpoor, 2016). A signature-based approach with two IDS engines, Suricata and Snort is described by (Waagsnes, 2017) for IEC 60870-5-104, Distributed Network Protocol 3 (DNP3), Modbus, and MITM. The limitation of the signature based detection is that it only works and recognizes known threat patterns and would fail for a zero-day attack. The IDS signatures are well known to both the system and malicious users (Paterson, 2011) and thus the attackers can take time to circumvent it (Winn et al., 2015). Therefore it is critical to detect the intrusion patterns that have not been seen before.

The machine learning application to reduce the manual configurations for IDS is discussed by (Mantere, et al., 2013). An IDS to detect malicious traffic was described by (Maglaras, & Jiang, 2014) combining OCSVM with recursive K-means clustering to minimize false alarms. OCSVM was found to perform well in comparison to other methods for anomaly detection; however, the performance could degrade for higher dimensional problems (Sander, 2017). Machine learning techniques have been extensively used but the recent breakthrough has been in neural networks.

An in-vehicle network security IDS using deep learning was proposed for Controller Area Network (CAN) traffic by (Kang, & Kang, 2016) addressing vanishing gradient problems through Deep Belief Networks (DBN). Anomaly detection is regarded as an unsupervised task and a semi-supervised scheme utilizing some labelled data was proposed by (Görnitz, et al., 2013). LSTM for collective anomaly detection for network traffic was proposed by (Thi et al., 2017) for KDD 1999 (KDD Cup 1999 Data). Recurrent Neural Network (RNN) work well with temporal data and store input event representations using their feedback connections (Hochreiter, & Schmidhuber, 1997). Tuor, et al. (2017) proposed an unsupervised deep learning approach based on RNN to characterize users to defend against insider threat using system logs. User behaviour through web log files was investigated by (Ma, et al., 2017) using Multiple Layer Perceptron (MLP) and Decision Trees. A joint model of LSTM and OCSVM (or SVDD) was trained for variable length data sequences by (Ergen, 2017) and the techniques also applied to Gated Recurrent Unit (GRU).

Autoencoders

Autoencoders are neural networks that can be unsupervised and work by creating a compressed representation of its input data (Figure 2). The input and output layers have the same dimensions whereas the hidden layer is a lower dimensional or compressed representation. It comprises of an encoder and decoder. The encoder generates a low dimensional or compressed representation from the

original high dimensional data, whereas the decoder uses the compressed representation and expands it to the same higher dimensional data as the original (Alla, 2019). In the process of encoding to a low dimension, the encoder discards irrelevant data and learns important features (Alla, 2019). Ideally we would have a low reconstruction loss between the input and reconstructed output from the compressed representation. Thus the reconstructed or expanded output is only an approximation of the input. The difference in approximating the input can be measured as a reconstruction error or loss. The reconstruction loss will be low for inputs similar to the ones used to train the autoencoder but would be higher for dissimilar inputs.

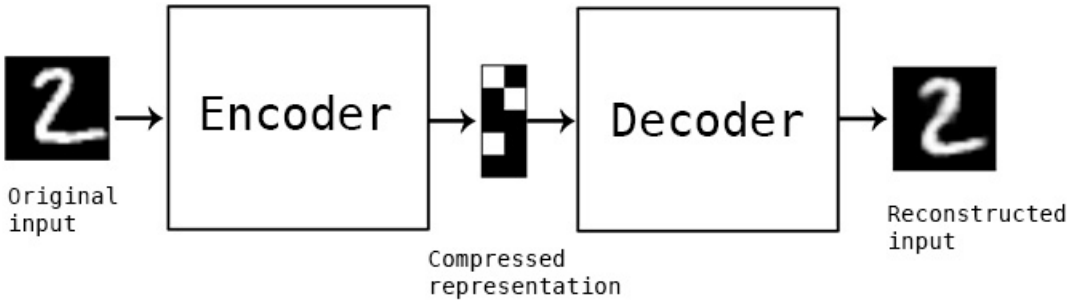

Figure 2. Relationship between original, compressed and reconstructed input (Chollet 2016)

The autoencoders are especially interesting for anomaly detection based on the reconstruction loss. By measuring the reconstruction loss L (Equation 1, where X is the original input and $\hat{X}$ is the reconstruction input), or the information loss between the input and reconstructed output, it is possible to isolate an anomalous input from normal inputs. The reconstruction loss will be higher in case of an anomalous input.

$$L(X, \hat{X}) = \|X - \hat{X}\|^2 \quad (1)$$

The common applications of an autoencoder are segmentation, dimensionality reduction, and detection. Dimensionality reduction of data with multilayer neural networks using a small central layer to reconstruct the high dimensional input vectors was shown to produce better results than PCA **(**Hinton, G. E., & Salakhutdinov, 2006). The data projections learnt by autoencoders are more interesting than PCA or similar simple techniques (Chollet, 2016).

METHODS AND MATERIALS

Dataset

The gas pipeline dataset (Beaver, et al., 2013) used for this research, was specifically developed using an in-house SCADA gas pipeline to gather information in the network between a RTU and MTU, and also to provide a public SCADA dataset for IDS research. The data is provided in raw and Attribute-Relation File Format (ARFF) formats with a total of 274,627 instances in each dataset, with each row containing multiple columns or features (Turnipseed, 2015). Specifically, gas pipeline 5 dataset in ARFF format was used (Table 2). The percentage of normal instances is 78.1% and that of attack instances is 21.9%, thus the data is unbalanced. This would in general be true for any dataset as the proportion of the normal instances would be more than the attack instances.

Table 2. Features of Gas Pipeline Dataset (Turnipseed, 2015; Morris, 2015)

Feature	Feature Type	Description
Address	Network	The station address of the MODBUS slave device. This address is the same on a query and response to a given slave device
Function	Command Payload	MODBUS function code.
Length	Network	The length of the MODBUS packet
Setpoint	Command Payload	The pressure set point when the system is in the Automatic system mode.
Gain	Command Payload	PID gain
Reset rate	Command Payload	PID reset rate.
Deadband	Command Payload	PID dead band.
Cycle time	Command Payload	PID cycle time
Rate	Command Payload	PID rate.
System Mode	Command Payload	The system's mode automatic (2), manual (1), or off (0).
Control scheme	Command Payload	The control scheme is either pump (0) or solenoid (1). This determines which mechanism is used to regulate the set point.
Pump	Command Payload	Pump control; on (1) or off (0). Only used in manual mode.
Solenoid	Command Payload	Relief valve control; opened (1) or closed (0). Only used in manual mode.
Pressure measurement	Response Payload	Pressure measurement.
CRC rate	Network	Cyclic-Redundant Checksum rate
Command response	Network	Command (1) or response (0).
Time	Network	Time stamp.
Binary Attack	Label	Binary class; attack (1) or normal (0).
Categorized attack	Label	Category of attack (0-7).
Specific attack	Label	Specific attack (0-35)

The dataset has three types of features (Table 2), that is, network, payload and labels, and comprises of network information that can be used to train IDS (Turnipseed, 2015). Also, for the features Setpoint, gain, Reset rate, Deadband, Cycle time, Rate, System mode, Control scheme, Pump, Solenoid, and Pressure measurement there are a lot of missing values. Payload information shows the state, settings and parameters and can be used to detect if the system is in a critical state. Labels (Binary attack, Categorized attack, Specific attack) are used to indicate the normal or attack transactions.

Table 3. Attack Categorization (Turnipseed, 2015)

Type of Attack	Code	Threat Type
Normal	Normal(0)	N/A
Naïve Malicious Response Injection	NMRI(1)	Modification/Fabrication
Complex Malicious Response Injection	CMRI(2)	Modification/Fabrication
Malicious State Command Injection	MSCI(3)	Modification/Fabrication
Malicious Parameter Command Injection	MPCI(4)	Modification/Fabrication
Malicious Function Code Injection	MFCI(5)	Modification/Fabrication
Denial of Service	DoS(6)	Interruption
Reconnaissance	Recon(7)	Interception

Further details about the dataset are available in (Turnipseed, 2015; Morris et al., 2015). The authors (Turnipseed, 2015; Morris et al., 2015) also provide information about missing data which has occurred due to Modbus frames being represented as rows which means that information in each row is not the same and some features are undefined.

Anomaly Detection Techniques

Multiclass Classification

We used two of the unsupervised machine learning techniques, that is, K-nearest neighbours (KNN) and Support Vector Machines (SVM) for multiclass classification in order to provide comparison against other reported results in the literature for the gas pipeline dataset. KNN classifier (Scikit-learn) is an unsupervised scheme that can cluster the data based on a distance measure. Support Vector Classification (SVC) (based on libsvm) by default uses a one-vs-one strategy for multiclass classification.

Binary Classification

For binary classification, we used autoencoder implementations using Keras (Keras) with Tensorflow (Abadi et al., 2016) as backend. Keras is an open source neural networks library enabling quick implementation (Keras). In order to train the autoencoder, we used the same split of training and test data as used across other techniques for ease of comparison.

Performance Metrics

The performance of anomaly detection is based on how well the model performs on the unseen data. It is common practice to consider the evaluation results of the model into four categories: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) (Figure 3).

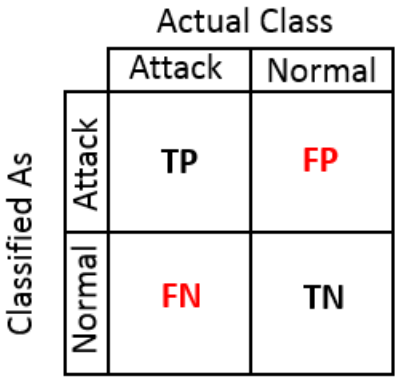

Figure 3. Four possible classifier output categories

Figure 3 shows the outputs of a classifier categorized into four classes. Two of the diagonal outcomes shown in black (TP and TN) are good, as they represent correct classifications. TP here denotes an actual attack classified as an attack, whereas TN is a normal instance classified as a normal type. A FP is a classification of a normal instance as an attack type, whereas FN is a classification of an attack as a normal instance, both being cases of misclassifications.

We use the following metrics (Sokolova, & Lapalme, 2009; Tharwat, 2018) for the evaluation of our models:

- *Classification Accuracy*
  This indicates the accuracy of the classifier in making a correct decision. It is expressed as:
  $CA = (TP + TN)/(TP + TN + FP + FN)$ (2)

- *Precision*
  For a positive predicted value, it indicates the percentage of correct predictions. This is expressed as:
  $Precision = TP/(TP + FP)$ (3)

- *Recall*
  For an actual positive value, it indicates the percentage of correct predictions. It is expressed as:
  $Recall = TP/(TP + FN)$ (4)

- *F1 score*
  $F1 = (2XPrecisonXRecall)/(Precision + Recall)$ (5)

RESULTS AND DISCUSSION

Pre-processing and Feature Selection

The data pre-processing and feature extraction is an important first step in data analysis (Tuor, et al., 2017; Ma, et al., 2017). The chosen dataset has many undefined values across six features represented with a '?'. After considering the range of known values, we changed all the '?' values to -1 in order to apply the classification algorithms. We found this did not affect the results negatively when comparing the results with the reduced subset based on deleting columns with '?' values or that based on RFE selection.

A scatter plot depicting the spread of PCA transformed values for normal and seven attack type instances is shown in Figure 4. It can be seen that there is some overlap in different attack categories, e.g., MPCI and DoS as shown in the top right corner. This is important to note for interpreting the results later. Due to the overlap, it becomes difficult to fully segregate some categories.

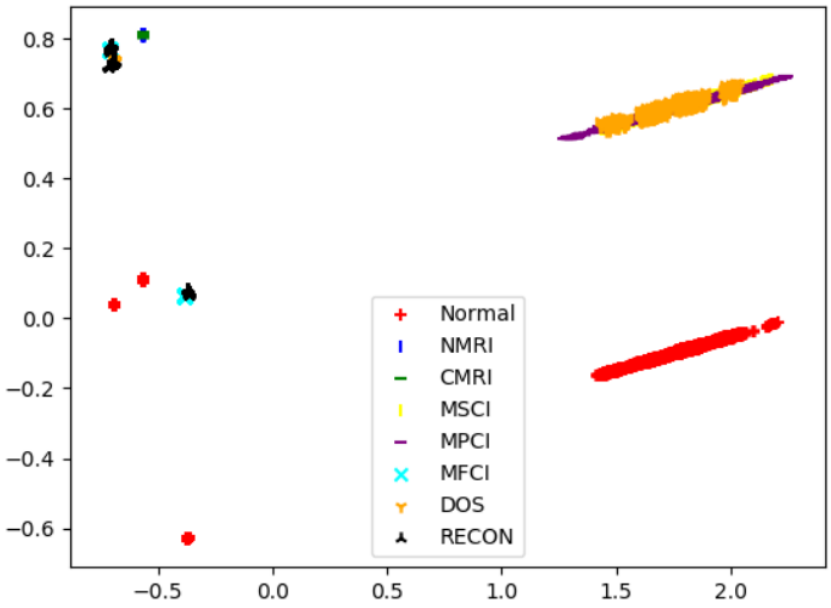

Figure 4. PCA normalized values for normal data and the seven attack categories

We also chose a reduced feature subset by applying RFE to the dataset and selected 10 of the most relevant features. However, this only improved the KNN results. The reason for this could be the hidden relationships between features.

The attack instances in the dataset are less compared to the normal instances and the dataset can be considered unbalanced, therefore we applied the models on the full dataset but also used class weighting, which is available in different machine learning software.

Anomaly Detection - Multiclass Classification

We applied two techniques for multiclass anomaly classification. The results for the two selected models, that is KNN and SVM are provided in Table 4 along with the results from (Shirazi, et al., 2016; Feng, et al., 2017). The best results across all techniques are shown in bold. The results in (Shirazi, et al., 2016) utilize K-means, NB, PCA-SVD and GMM and the best of the results for each category are shown in Table 4. However, it is important to consider that Shirazi, et al. (2016) created data subsets by combining each of the seven anomalous classes with normal data. In comparison we used the machine learning models over the single combined dataset with all eight classes.

Table 4. Multiclass Classification Results on Gas Pipeline Dataset

Attack	Technique	Precision	Recall	F-Score
NMRI	KNN	0.64	0.60	**1.0**
	SVC	0.37	**0.99**	0.53
	(Shirazi, et al., 2016)	**1.0 (NB, GMM)**	0.81 (NB)	0.87 (NB)
	(Feng et al., 2017)	-	0.88	-
CMRI	KNN	0.77	0.80	0.62
	SVC	0.97	0.022	0.04
	(Shirazi, et al., 2016)	**1.0 (NB)**	**0.83 (NB)**	**0.87 (NB)**
	(Feng et al., 2017)	-	0.67	-
MSCI	KNN	0.91	**0.99**	0.79
	SVC	0.64	0.43	0.52
	(Shirazi, et al., 2016)	**0.99 (PCA-SVD)**	0.72 (NB, K-means)	**0.82 (NB, K-means)**
	(Feng et al., 2017)	-	0.62	-
MPCI	KNN	**0.98**	**0.97**	**0.95**
	SVC	0.77	0.91	0.83
	(Shirazi, et al., 2016)	0.70 (NB)	0.66 (NB)	0.74 (GMM)
	(Feng et al., 2017)	-	0.80	-
MFCI	KNN	0.95	**1.0**	**0.97**
	SVC	0.91	**1.0**	0.95
	(Shirazi, et al., 2016)	**1.0 (NB, GMM)**	0.54 (PCA-SVD)	0.67 (PCA-SVD)
	(Feng et al., 2017)	-	**1.0**	-
DoS	KNN	**1.0**	0.89	**0.94**
	SVC	**1.0**	0.40	0.58
	(Shirazi, et al., 2016)	**1.0 (K-means)**	0.79 (NB)	0.88 (NB)
	(Feng et al., 2017)	-	**0.94**	-
Reconnaissance	KNN	**1.0**	0.93	**0.96**
	SVC	0.99	0.87	0.92
	(Shirazi, et al., 2016)	0.99 (NB)	0.75 (K-means)	0.84 (K-means)
	(Feng et al., 2017)	-	**1.0**	-

There is no clear best algorithm but in general the results for KNN and NB are better across many cases. Our proposed architecture does not require steps such as noise addition and additional labels. Missing data handling is not discussed by (Shirazi, et al., 2016; Feng et. al, 2017).

Anomaly Detection - Binary Classification

Classification

We used the autoencoder for binary classification of the data into attack and normal instances, and provide comparison of results with (Shirazi, et al., 2016; Feng, et al., 2017). In order to segregate the attack instances from the normal instances we require a suitable threshold on reconstruction loss.

A plot of TP, TN, FP and FN by selecting reconstruction loss threshold values from 0 to 2.5 is shown in Figure 5. An appropriate threshold value can identify anomalies based on their loss value being higher compared to the normal instances. As shown in Figure 5, if the threshold is chosen as too low (less than around 0.3) then some normal instances will be categorised as attack (FP), and similarly if it is too high (greater than around 1.1) then some attack instances will be categorised as normal (FN). We provide results for two selections of threshold as 0.3 and 1.8.

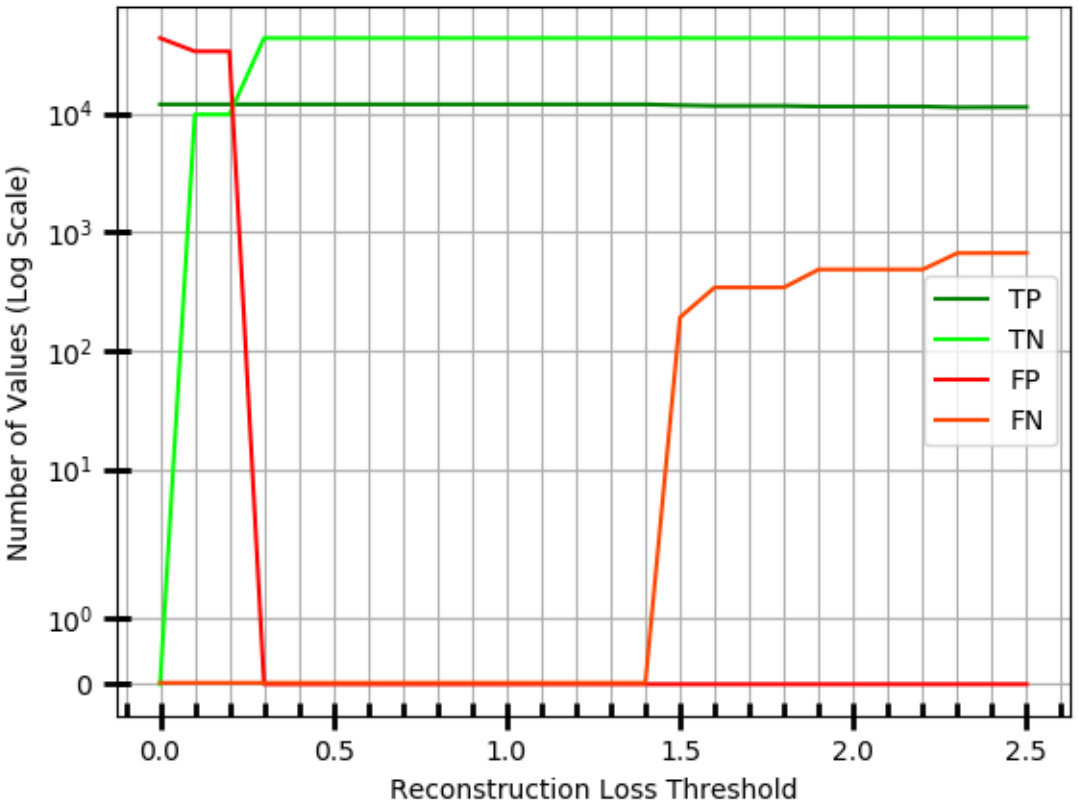

Figure 5. The values of TP, TN, FN, and FP corresponding to a range of reconstruction loss thresholds

Comparative results for other techniques reported in the literature with the proposed autoencoder based scheme (with a loss threshold of 0.3 and 1.8) are shown in Table 5. Feng, et al. (2017) combine a Bloom filter to store the signature data base and LSTM network based softmax classifier. The data is pre-processed to segregate the normal data which is further fragmented into packages based on their time series anomaly detector. The model is based on selection of a parameter λ and k. The value of k =

4 provided highest F-scores. Adding probabilistic noise during training is proposed to avoid false positives. They provide comparisons with BF, BN, SVDD, IF, GMM and PCA-SVD by tuning hyper parameters (Nazir et al.2018) for best F-scores. (Shirazi et al, 2016) provide results of four techniques (K-means, PCA-SVDD, NB, GMM of which the best results are represented for each category in Table 5. Mansouri, et al. (2017) used a set of 24 techniques on the dataset and reported an accuracy of 97.5% for their proposed system based on a Gray Wolf Optimizer (GWO) algorithm.

It can be seen from Table 5 that the results for the proposed autoencoder based scheme for both a low and high threshold provide improved results compared to other techniques. Anomaly detection based on autoencoders only requires the setting of a threshold. After setting the threshold, the system can autonomously determine and report any anomalous data. With a threshold value of 1.8, although it gives good results, it also allows some attack values to be passed undetected. Similarly, considering threshold value of 0.3 (Figure 5), any decrease below this has a progressive increase in the number of FP, from a very low to a very high value. The important fact to note here is that unlike many other schemes where there are very tight constraints for the threshold value, an optimum threshold value exists as a continuum from 0.3 to 1.0, where FN and FP values do not occur.

Table 5. Comparative results for Anomaly Detection using Autoencoder (Normal and Attack Traffic) with a reconstruction loss threshold of 1.0 and 1.8

Technique	**Precision**	**Recall**	**Accuracy**	**F-Score**
Autoencoder (threshold = 0.3)	1.0	1.0	1.0	1.0
Autoencoder (threshold = 1.8)	1.0	0.97	0.99	0.98
LSTM (Feng et al., 2017)	0.94	0.78	0.92	0.85
K-means, NB, PCA-SVD, GMM (Shirazi et al., 2016)	0.8319 (K-means)	0.7692 (NB)	0.9036 (NB)	0.8605 (NB)
GWO (Mansouri et al., 2017)	-	-	0.975	-

There is an obvious trade-off in the choice of threshold that depends on the application type. For a SCADA network, a high threshold will result in FN values, that is, some attack patterns will pass through as normal data, which could be disastrous. Choice of a lower threshold will suppress FN but depending on the traffic may result in occasional FPs which is better than having FNs for critical applications. We contend that a suitable threshold can be selected based on the available normal data patterns to then differentiate between an attack and a normal instance of network traffic.

Determining the Threshold Value

The training of autoencoder with only the normal data is sufficient to learn input features and a corresponding reconstruction loss for normal instances. Therefore it is useful to select a threshold value based on the reconstruction loss distribution. The distribution of reconstruction loss for the test data (Figure 5) is shown in Figure 6.

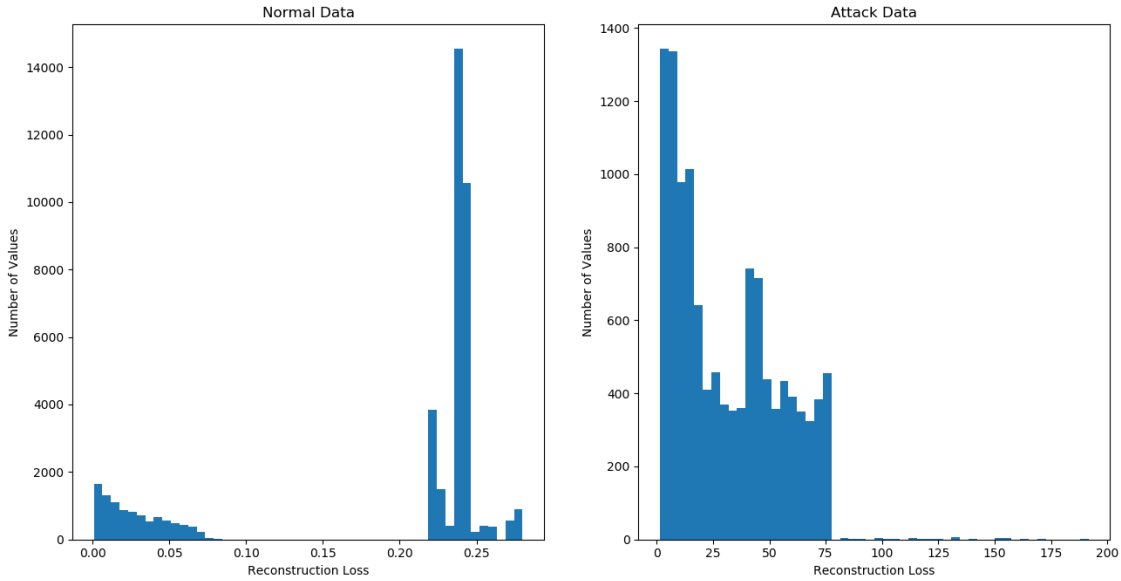

Figure 6. Histogram showing the number of values vs reconstruction threshold for (a) Normal data (b) Attack data

From Figure 6, it can be seen that the normal data has a maximum reconstruction loss of approximately 0.28. A threshold value greater than this will ensure there are no FPs. Similarly, the minimum reconstruction loss for attack data is approximately 1.0. A threshold value less than 1.0 ensures that there are no FNs. Further data exploration and statistical techniques can help where there is an overlap of an attack and normal reconstruction loss values.

Discussion

Intrusion detection for SCADA systems has become an active and major research area due to the current cyber threat environment and the high potential to damage because the systems can be easily compromised. Anomaly detection is an unsupervised learning task as the anomalous distributions are unknown. Cyber security research is focusing on unsupervised neural networks based classification methodologies that promise good results and are capable of handling complex systems in near real-time.

In actual practice the attack or anomalous instances will always be less than the normal instances thus creating unbalanced datasets. This can create problems for supervised learning techniques that require labelled datasets from each type for training. This can be overcome by using either data augmentation of the attack instances or by assigning class weights to the classes with less representation.

However, progress in developing new unsupervised algorithms that can provide better detections with a low false alarm rate, is required. Timely detection of an anomaly is an important characteristic of IDS. Therefore, in this study, we used autoencoders to propose a threshold based scheme that can quickly discern an attack instance. The autoencoder reconstruction loss can help segregate an anomalous input from the normal traffic pattern. This segregation would be based on a threshold of reconstruction error. Autoencoders provide better results compared to other techniques. An added

advantage is that these require only normal traffic for training, and thereafter the deviation of an attack pattern can be determined easily. Thus the choice of threshold will determine the number of inputs that get flagged as an anomaly and therefore thresholds have to be carefully chosen.

As seen in Figure 5, threshold is not a very rigid or fixed value but rather it comprises of a range of values that will give optimum results. The threshold can be kept low enough so that no attack data (even if it is very similar to normal traffic) can pass through. A threshold below 0.3 will have some FP values, and similarly a threshold higher than around 1.0 will have some FN values. For SCADA applications a false alarm for a FP value would be deemed better than a real attack being passed undetected as a FN.

The model and the threshold can also be tailored with the passage of time. The autoencoder model can continuously learn and improve after encountering more traffic patterns and the threshold can also change based on the desired outcomes and constraints. We contend that for timely and accurate unsupervised anomaly detection in critical SCADA networks, it is sufficient to identify an attack instance based on binary classification only.

The availability of public SCADA datasets has been addressed only to a limited extent. However, there is still a need to validate research results for network traces from the real-world SCADA systems. This can be helped by the fact that some SCADA vendors have started to provide the machine learning and data analytics packages within their SCADA products (WinCC OA SmartSCADA).

CONCLUSION

The integration of SCADA network and devices with the Internet has many business benefits. However, it has also made these otherwise highly secure and critical systems, vulnerable to attackers, who can potentially access and exploit the intrinsic weaknesses in many of the access protocols.

This paper investigated the application of autoencoder neural networks on a realistic SCADA laboratory dataset for anomaly detection system for SCADA system. The results show that the proposed autoencoder based techniques can uncover anomalies in the data with a high degree of probability. Thus in an autoencoder based unsupervised technique, the normal traffic features are learned and an attack pattern can be detected as an anomaly based only on its reconstruction loss or distance from the normal dataset. The use of autoencoders is especially promising for anomaly detection as they can be trained using only the normal SCADA data (which is available in abundance and easily acquired), thereafter detecting an unseen attack instance based on its reconstruction loss or distance from the normal data. We also used KNN and SVM to the gas pipeline SCADA dataset for multiclass classification to compare with other results in research literature.

The provision of an integrated data analysis package within the SCADA application is becoming important with some vendors starting to provide such facility. For future work, we will consider the data analysis packages integrated within SCADA software in order to develop a fully integrated anomaly detection system.

ACKNOWLEDGMENTS

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

The authors want to thank T. Morris and the Mississippi State University for making available a public SCADA dataset.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B. Tucker, P., Vasudevan, V., Warden, P., & Zheng X. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, Savannah, GA, 265-283.

Ahmed, S., Lee, Y., Hyun, S., & Koo, I. (2019). Mitigating the impacts of covert cyber attacks in smart grids via reconstruction of measurement data utilizing deep denoising autoencoders. *MDPI Energies*, 12(16), 3091.

Alla, S., & Adari, S. K. (2019). Beginning anomaly detection using Python-based deep learning. Apress.

Almalawi, A., Yu, X., Tari, Z., Fahad, A., & Khalil, I. (2014). An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems. *Computers and Security*, 46, 94-110.

Aygun, R. C., & Yavuz, A. G. (2017). Network anomaly detection with stochastically improved autoencoder based models. *IEEE 4th International Conference on Cyber Security and Cloud Computing.*

Beaver, J. M., Borges-Hink, R. C., & Buckner, M. A. (2013). An evaluation of machine learning methods to detect malicious SCADA communications. *In Proceedings of 12th International Conference on Machine Learning and Applications (ICMLA),* (vol. 2, pp.54-59).

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Survey & Tutorials*, 18(2), 1153-1176.

Chollet, F. (2016). Building autoencoders in Keras. In tutorials, The Keras Blog. https://blog.keras.io/building-autoencoders-in-keras.html

Cyber security breaches survey. (2017). Department for Digital, Culture, Media & Sport, 19 April 2017. https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2017.

Digital Bond: Peterson D. (2011). Quickdraw IDS 4.1 Release. http://www.digitalbond.com/blog/2011/02/28/quickdraw-ids-4-1-release/

Ergen, T., Mirza, A. H., & Kozat. S. S. (2017). Unsupervised and semi-supervised anomaly detection with LSTM neural networks. arXiv:1710.09207 [eess.SP].

Erol-Kantarci, M., & Mouftah, H. T. (2013). Smart grid forensic science: applications, challenges, and open issues. *IEEE Commun Magaz*, 51(1), 68-74.

Fan, C., Xiao, F., Zhao, Y., & Wang, J. (2018). Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. *Applied Energy*, 211, 1123-1135.

Feng, C., Li, T., & Chana, D. (2017). Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks. *47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Denver, CO.

Genge, B., Siaterlis, C., Fovino, I. N., & Masera, M. (2012). A cyber-physical experimentation environment for the security analysis of networked industrial control systems. *Comput Electr Eng*, 38(5), 1146-1161.

Görnitz, N., Kloft, M., Rieck, K., & Brefeld, U. (2013).Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46(1), 235-262.

Heenan, R., & Moradpoor, N. (2016). *A survey of intrusion detection system technologies*. PGCS 2016: The First Post Graduate Cyber Security Symposium, Edinburgh Napier University, UK.

Hinton, G. E., & Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313, 504–507.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.

Igure, V. M., Laughter, S. A., & Williams, R. D. (2006). Security issues in SCADA networks. *Comput Secur*, 25(7), pp. 498-506.

Jain, P., & Tripathi, P. (2013). SCADA security: a review and enhancement for DNP3 based systems *CSIT*, 1(4), 301-308.

Jiang, J., & Yasakethu, L. (2013). Anomaly detection via one class SVM for protection of SCADA systems. *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 82-88.

Kang, Min-Joo, & Kang, Je-Won. (2016). Intrusion detection system using deep neural network for in-vehicle network security. *PLoS ONE*, 11(6).

Keras. [Software] https://keras.io/

KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Langner, R. (2011). Stuxnet-Dissecting a cyberwarfare weapon. *IEEE Secur Privacy*, 9(3), 49–51.

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.

Liu, L., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11-26.

Lutz, M, Vogt, S., Berkhout, V., Faulstich, S., Dienst, S., Steinmetz, U., Guck, C., & Ortega, A. (2020). Evaluation of anomaly detection of an autoencoder based on maintenance information and SCADA data. *MDPI Energies*, 13(5), 1-18.

Ma, K., Jiang, R., Dong, M., Jia, Y., & Li, A. (2017). *Neural network based web log analysis for web intrusion detection*. In: Wang G., Atiquzzaman M., Yan Z., Choo KK. (eds) Security, Privacy, and Anonymity in Computation, Communication, and Storage. SpaCCS 2017. Lecture Notes in Computer Science, 10658. Springer, Cham.

Maglaras, L. A., & Jiang, J. (2014). A real time OCSVM intrusion detection module with low overhead for SCADA systems. *International Journal of Advanced Research in Artificial Intelligence*, 3(10).

Mantere, M., Sailio, M., & Noponen, S. (2013). Network traffic features for anomaly detection in specific industrial control system network. *MDPI Future Internet*, 5(4), 460-473.

Mansouri, A., Majidi, B., & Shamisa, A. (2017). Anomaly detection in industrial control systems using evolutionary-based optimization of neural networks. *Communications on Advanced Computational Science with Applications*, 2017(1), 49-55.

Morris, T. H., Thornton, Z., & Turnipseed, I. (2015). Industrial control system simulation and data logging for intrusion detection system research. *7th Annual Southeastern Cyber Security Summit*, Huntsvile, AL.

Nader, P., Honeine, P., & Beauseroy, P. (2014). lp-norms in one-class classification for intrusion detection in SCADA systems. *IEEE Trans. on Industrial Informatics*, 10(4).

Nazir, S., Patel, S., & Patel, D. (2017). Assessing and augmenting SCADA cyber security: A survey of techniques. *Elsevier Computers and Security*, 70, 436-454.

Nazir, S., Patel, S., & Patel, D. (2018). Hyper parameters selection for image classification in convolutional neural networks. *17th IEEE International Conference on Cognitive Informatics and Cognitive Computing*, Berkeley, CA, pp. 401-407.

Sander, S. (2017). *Predictive maintenance using machine learning methods in petrochemical refineries*. [MS thesis, Delft University of Technology]. http://resolver.tudelft.nl/uuid:e95f39a4-569a-470e-a431-962b9766a302

Schuster, F., Paul, A., Rietz, R., & Konig, H. (2015). Potentials of using one-class SVM for detecting protocol-specific anomalies in industrial networks. *IEEE Symposium Series on Computational Intelligence*, Cape Town, 83-90.

Scikit-learn: [Software] http://scikit-learn.org/stable/index.html.

Shirazi, S. N., Gouglidis, A., Syeda, K. N., Simpson, S., Mauthe, A., Stephanakis, I. M., & Hutchison, D. (2016). Evaluation of anomaly detection techniques for SCADA communication resilience. 2016 Resilience Week (RWS), Chicago, IL, 140-145.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437.

Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics.* https://doi.org/10.1016/j.aci.2018.08.003

Thi, N. N., Cao, V. L., & Le-Khac, N. (2017). One-class collective anomaly detection based on LSTM-RNNs. *Transactions on Large-Scale Data and Knowledge-Centered Systems XXXVI*, Springer Berlin Heidelberg, 2017.

Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson S. (2017). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. arXiv:1710.00811 [cs.NE].

Turnipseed, I. (2015). *A new SCADA dataset for intrusion detection system research*. [MS thesis, Mississippi State University]. *http://sun.library.msstate.edu/ETD-db/theses/available/etd-06292015-115535/*

Waagsnes, H. (2017). *SCADA Intrusion Detection System Test Framework*. [MS thesis, University of Agder]. http://hdl.handle.net/11250/2455016

Wilson, D., Tang, Y., Yan, J., & Lu, Z. (2018). Deep learning-aided cyber-attack detection in power transmission systems. *IEEE Power & Energy Society General Meeting (PESGM)*, Portland, OR, 1-5.

WinCC OA SmartSCADA. https://new.siemens.com/global/en/products/automation/industry-software/automation-software/scada/simatic-wincc-oa/wincc-oa-options.html

Winn, M., Rice, M., Dunlap, S., Lopez, J., & Mullins, B. (2015). Constructing cost effective and targetable industrial control system honeypots for production networks. *Elsevier Int J Crit Infrastr Protect*, 10, 47-58.

Yousefi-Azar, M., Varadharajan, V., Hamey, L., & Tupakula, U. (2017). Autoencoder-based feature learning for cyber security applications. *International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, 3854-3861.

Zhu, B., Joseph, A., & Sastry, S. (2011). A taxonomy of cyber attacks on SCADA systems. *International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, Dalian, 380-388.